

## EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	420	(715/511).CCLS.	US-PGPUB; USPAT	OR	OFF	2006/11/06 13:30
L2	1237	(715/530).CCLS.	US-PGPUB; USPAT	OR	OFF	2006/11/06 13:30
L3	203	(715/509).CCLS.	US-PGPUB; USPAT	OR	OFF	2006/11/06 13:30
L4	1615	(707/203).CCLS.	US-PGPUB; USPAT	OR	OFF	2006/11/06 13:31
L5	64	first with table with new with version	US-PGPUB; USPAT	OR	ON	2006/11/06 13:36
L6	12	second with table with old\$4 with version	US-PGPUB; USPAT	OR	ON	2006/11/06 13:37
L7	70	first with table with new\$4 with version	US-PGPUB; USPAT	OR	ON	2006/11/06 13:37
L8	8	6 and 7	US-PGPUB; USPAT	OR	ON	2006/11/06 13:37
L9	611	(database or table) with old\$4 with version	US-PGPUB; USPAT	OR	ON	2006/11/06 13:37
L10	2315	(database or table) with new\$4 with version	US-PGPUB; USPAT	OR	ON	2006/11/06 13:37
L11	2545	9 or 10	US-PGPUB; USPAT	OR	ON	2006/11/06 13:37
L12	381	9 and 10	US-PGPUB; USPAT	OR	ON	2006/11/06 13:37
L13	381	8 or 12	US-PGPUB; USPAT	OR	ON	2006/11/06 13:37
L14	184	13 and (version with identif\$7)	US-PGPUB; USPAT	OR	ON	2006/11/06 13:38
L15	172	14 and (updat\$4)	US-PGPUB; USPAT	OR	ON	2006/11/06 13:38
L16	143	15 and (updat\$4 with version)	US-PGPUB; USPAT	OR	ON	2006/11/06 13:40
L17	660	sql with column with query	US-PGPUB; USPAT	OR	ON	2006/11/06 13:40
L18	8557	obtain\$4 adj2 column	US-PGPUB; USPAT	OR	ON	2006/11/06 13:40
L19	25	L17 and L18	US-PGPUB; USPAT	OR	ON	2006/11/06 13:40
L20	315	sql with advantage	US-PGPUB; USPAT	OR	ON	2006/11/06 13:40
L21	2	L19 and L20	US-PGPUB; USPAT	OR	ON	2006/11/06 13:40

## EAST Search History

L22	124	16 and (search\$4 or quer\$4)	US-PGPUB; USPAT	OR	ON	2006/11/06 13:40
L23	200	database with old with version	US-PGPUB; USPAT	OR	ON	2006/11/06 13:40
L24	1138	database with new with version	US-PGPUB; USPAT	OR	ON	2006/11/06 13:40
L25	124	L23 and L24	US-PGPUB; USPAT	OR	ON	2006/11/06 13:40
L26	30308	version with number	US-PGPUB; USPAT	OR	ON	2006/11/06 13:40
L27	74	L25 and L26	US-PGPUB; USPAT	OR	ON	2006/11/06 13:40
L28	50	L27 and (version with value)	US-PGPUB; USPAT	OR	ON	2006/11/06 13:41

	Document ID	Issue Date	Title	Current OR
1	US 20060212487 A1	20060921	METHODS AND SYSTEMS FOR MONITORING TRANSACTION ENTITY VERSIONS FOR POLICY COMPLIANCE	707/200
2	US 20060212486 A1	20060921	METHODS AND SYSTEMS FOR COMPLIANCE MONITORING KNOWLEDGE BASE	707/200
3	US 20060053481 A1	20060309	System and method for ensuring secure transfer of a document from a client of a network to a printer	726/3
4	US 20060036656 A1	20060216	Automatic versioning and data mutation of user- defined data types	707/203
5	US 20050278579 A1	20051215	System and method for building a run- time image from components of a software program	714/38
6	US 20050257211 A1	20051117	Method and mechanism for managing incompatible changes in a distributed system	717/170
7	US 20050216530 A1	20050929	Method and apparatus for updating a stored version of content stored in a storage device	707/203
8	US 20050209876 A1	20050922	Methods and systems for transaction compliance monitoring	705/1

	Document ID	Issue Date	Title	Current OR
9	US 20050165760 A1	20050728	Auto version managing system and method for use in software	707/3
10	US 20040260607 A1	20041223	Stored product personal identification system	705/14
11	US 20040230952 A1	20041118	Marking changes based on a region and a threshold	717/120
12	US 20040153469 A1	20040805	Database comparator	707/101
13	US 20040024765 A1	20040205	System and method for auditing and reconciliation of telephone service providers' databases with a regional or central data repository	707/10
14	US 20030121024 A1	20030626	System and method for building a runtime image from components of a software program	717/107
15	US 20030120688 A1	20030626	Versioning model for software program development	707/203
16	US 20030120678 A1	20030626	Prototyping model for components of a software program	707/102
17	US 20020161753 A1	20021031	Distributed document retrieval method and device, and distributed document retrieval program and recording medium recording the program	707/3

	Document ID	Issue Date	Title	Current OR
18	US 20020103815 A1	20020801	High speed data updates implemented in an information storage and retrieval system	707/203
19	US 20020073110 A1	20020613	Version collection technique implemented on an intrinsic versioning information storage and retrieval system	707/206
20	US 7047242 B1	20060516	Weighted term ranking for on-line query tool	707/10
21	US 6826559 B1	20041130	Hybrid category mapping for on-line query tool	707/3
22	US 6643640 B1	20031104	Method for performing a data query	707/3
23	US 6636868 B1	20031021	System and method for auditing and reconciliation of telephone service provider's databases with a regional or central data repository	707/104.1
24	US 6578056 B1	20030610	Efficient data transfer mechanism for synchronization of multi-media databases	715/500.1
25	US 6564232 B1	20030513	Method and apparatus for managing distribution of change-controlled data items in a distributed data processing system	707/203

	Document ID	Issue Date	Title	Current OR
26	US 6519592 B1	20030211	Method for using data from a data query cache	707/6
27	US 6496843 B1	20021217	Generic object for rapid integration of data changes	715/526
28	US 6493721 B1	20021210	Techniques for performing incremental data updates	707/104.1
29	US 6484161 B1	20021119	Method and system for performing online data queries in a distributed computer system	707/3
30	US 6473503 B1	20021029	System, method and computer program product for achieving local number portability network management support	379/221.13
31	US 6473502 B1	20021029	System, method and computer program product for achieving local number portability costing and network management support	379/221.13
32	US 6457065 B1	20020924	Transaction-scoped replication for distributed object systems	719/328
33	US 6421683 B1	20020716	Method and product for performing data transfer in a computer system	707/104.1
34	US 6415295 B1	20020702	Storing personal medical information	707/104.1

	Document ID	Issue Date	Title	Current OR
35	US 6415028 B1	20020702	System and method that allows a telephone data repository to communicate with an external system	379/221.13
36	US 6411698 B1	20020625	System and method for communication between a telephone data repository and downstream data processing applications	379/207.01
37	US 6408294 B1	20020618	Common term optimization	707/5
38	US 6397228 B1	20020528	Data enhancement techniques	707/203
39	US 6393415 B1	20020521	Adaptive partitioning techniques in performing query requests and request routing	707/2
40	US 6374241 B1	20020416	Data merging techniques	707/6
41	US 6370548 B1	20020409	System and method for achieving local number portability	707/204
42	US 6298478 B1	20011002	Technique for managing enterprise JavaBeans (.TM.) which are the target of multiple concurrent and/or nested transactions	717/170
43	US 6088694 A	20000711	Continuous availability and efficient backup for externally referenced objects	707/8

	Document ID	Issue Date	Title	Current OR
44	US 6082776 A	20000704	Storing personal medical information	283/72
45	US 6067354 A	20000523	Method and system for processing data records from a telephone data repository to a receiving system	379/201.01
46	US 6047045 A	20000404	System and method for testing a telephone data interface system	379/26.01
47	US 5937343 A	19990810	Method and system for updating replicated databases in a telecommunication network system	455/403
48	US 5317731 A	19940531	Intelligent page store for concurrent and consistent access to a database by a transaction processor and a query processor	707/8
49	US 5280612 A	19940118	Multiple version database concurrency control system	707/8
50	US 5212789 A	19930518	Method and apparatus for updating application databases used in a distributed transaction processing environment	707/8




[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide


**THE ACM DIGITAL LIBRARY**

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used **tables store versions data update identifiers**

 Found **101,542** of **186,958**

Sort results by


[Save results to a Binder](#)

 Try an [Advanced Search](#)

Display results


[Search Tips](#)

 Try this search in [The ACM Guide](#)
☐ Open results in a new window

Results 1 - 20 of 200

 Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

 Relevance scale ☐ ☐ ☐ ☐ ☐

### 1 [Industrial sessions: beyond relational tables: Coordinating backup/recovery and data](#)



#### [consistency between database and file systems](#)

Suparna Bhattacharya, C. Mohan, Karen W. Brannon, Inderpal Narang, Hui-I Hsiao, Mahadevan Subramanian

 June 2002 **Proceedings of the 2002 ACM SIGMOD international conference on Management of data SIGMOD '02**

Publisher: ACM Press

 Full text available: [pdf\(1.44 MB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Managing a combined store consisting of database data and file data in a robust and consistent manner is a challenge for database systems and content management systems. In such a hybrid system, images, videos, engineering drawings, etc. are stored as files on a file server while meta-data referencing/indexing such files is created and stored in a relational database to take advantage of efficient search. In this paper we describe solutions for two potentially problematic aspects of such a data ...

**Keywords:** DB2, content management, database backup, database recovery, datalinks

### 2 [Garbage collection for a client-server persistent object store](#)



Laurent Amsaleg, Michael J. Franklin, Olivier Gruber

 August 1999 **ACM Transactions on Computer Systems (TOCS)**, Volume 17 Issue 3

Publisher: ACM Press

 Full text available: [pdf\(267.18 KB\)](#)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

We describe an efficient server-based algorithm for garbage collecting persistent object stores in a client-server environment. The algorithm is incremental and runs concurrently with client transactions. Unlike previous algorithms, it does not hold any transactional locks on data and does not require callbacks to clients. It is fault-tolerant, but performs very little logging. The algorithm has been designed to be integrated into existing systems, and therefore it works with standard i ...

**Keywords:** client-server system, logging, persistent object-store, recovery

### 3 [Distributed file systems: concepts and examples](#)




Eliezer Levy, Abraham Silberschatz

 December 1990 **ACM Computing Surveys (CSUR)**, Volume 22 Issue 4

Publisher: ACM Press

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

Full text available:  [pdf\(5.33 MB\)](#)[terms](#), [review](#)

The purpose of a distributed file system (DFS) is to allow users of physically distributed computers to share data and storage resources by using a common file system. A typical configuration for a DFS is a collection of workstations and mainframes connected by a local area network (LAN). A DFS is implemented as part of the operating system of each of the connected computers. This paper establishes a viewpoint that emphasizes the dispersed structure and decentralization of both data and con ...


#### 4 [Data replicas in distributed information services](#)



H. M. Gladney

March 1989 **ACM Transactions on Database Systems (TODS)**, Volume 14 Issue 1

Publisher: ACM Press

Full text available:  [pdf\(1.94 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [review](#)

In an information distribution network in which records are repeatedly read, it is cost-effective to keep read-only copies in work locations. This paper presents a method of updating replicas that need not be immediately synchronized with the source data or with each other. The method allows an arbitrary mapping from source records to replica records. It is fail-safe, maximizes workstation autonomy, and is well suited to a network with slow, unreliable, and/or expensive communications links ...

#### 5 [A Peer-to-Peer Replica Location Service Based on a Distributed Hash Table](#)

Min Cai, Ann Chervenak, Martin Frank

November 2004 **Proceedings of the 2004 ACM/IEEE conference on Supercomputing**

Publisher: IEEE Computer Society

Full text available:  [pdf\(343.25 KB\)](#) Additional Information: [full citation](#), [abstract](#)

A Replica Location Service (RLS) allows registration and discovery of data replicas. In earlier work, we proposed an RLS framework and described the performance and scalability of an RLS implementation in Globus Toolkit Version 3.0. In this paper, we present a Peer-to-Peer Replica Location Service (P-RLS) with properties of self-organization, fault-tolerance and improved scalability. P-RLS uses the Chord algorithm to self-organize PRLS servers and exploits the Chord overlay network to replicate ...

**Keywords:** Algorithms, Experimentation, Grid, Peer-to-Peer, Replication


#### 6 [The Kala basket: a semantic primitive unifying object transactions, access control, versions, and configurations](#)



Sergui S. Simmel, Ivan Godard

November 1991 **ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages, and applications OOPSLA '91**, Volume 26 Issue 11

Publisher: ACM Press

Full text available:  [pdf\(2.11 MB\)](#)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)


#### 7 [XML manipulations: Managing and querying multi-version XML data with update logging](#)



Raymond K. Wong, Nicole Lam

November 2002 **Proceedings of the 2002 ACM symposium on Document engineering**

Publisher: ACM Press

Full text available:  [pdf\(178.03 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

With the increasing popularity of storing content on the WWW and intranet in XML form, there arises the need for the control and management of this data. As this data is

constantly evolving, users want to be able to query previous versions, query changes in documents, as well as to retrieve a particular document version efficiently. This paper proposes a version management system for XML data that can manage and query changes in an effective and meaningful manner.

**Keywords:** XML, path expression, versioning

8 Status report of the graphic standards planning committee



Computer Graphics staff

August 1979 **ACM SIGGRAPH Computer Graphics**, Volume 13 Issue 3

**Publisher:** ACM Press

Full text available: pdf(15.01 MB) Additional Information: [full citation](#), [references](#), [citations](#)

9 Flexible update propagation for weakly consistent replication



Karin Petersen, Mike J. Spreitzer, Douglas B. Terry, Marvin M. Theimer, Alan J. Demers

October 1997 **ACM SIGOPS Operating Systems Review , Proceedings of the sixteenth ACM symposium on Operating systems principles SOSP '97**, Volume 31 Issue 5

**Publisher:** ACM Press

Full text available: pdf(2.16 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

10 The Desert environment



Steven P. Reiss

October 1999 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 8 Issue 4

**Publisher:** ACM Press

Full text available: pdf(868.64 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The Desert software engineering environment is a suite of tools developed to enhance programmer productivity through increased tool integration. It introduces an inexpensive form of data integration to provide additional tool capabilities and information sharing among tools, uses a common editor to give high-quality semantic feedback and to integrate different types of software artifacts, and builds virtual files on demand to address specific tasks. All this is done in an open and extensible ...

**Keywords:** integrated programming environments, program editors

11 A survey of peer-to-peer content distribution technologies



Stephanos Androutsellis-Theotokis, Diomidis Spinellis

December 2004 **ACM Computing Surveys (CSUR)**, Volume 36 Issue 4

**Publisher:** ACM Press

Full text available: pdf(517.77 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Distributed computer architectures labeled "peer-to-peer" are designed for the sharing of computer resources (content, storage, CPU cycles) by direct exchange, rather than requiring the intermediation or support of a centralized server or authority. Peer-to-peer architectures are characterized by their ability to adapt to failures and accommodate transient populations of nodes while maintaining acceptable connectivity and performance. Content distribution is an important peer-to-peer application ...

**Keywords:** Content distribution, DHT, DOLR, grid computing, p2p, peer-to-peer

**12** Multiversion-based view maintenance over distributed data sources

Songting Chen, Bin Liu, Elke A. Rundensteiner

December 2004 **ACM Transactions on Database Systems (TODS)**, Volume 29 Issue 4**Publisher:** ACM PressFull text available: [pdf\(480.72 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Materialized views can be maintained by submitting maintenance queries to the data sources. However, the query results may be erroneous due to concurrent source updates. State-of-the-art maintenance strategies typically apply compensations to resolve such conflicts and assume all source schemata remain stable over time. In a loosely coupled dynamic environment, the sources may autonomously change not only their data but also their schema or semantics. Consequently, either the maintenance or the ...

**Keywords:** View maintenance, transaction processing**13** Special system-oriented section: the best of SIGMOD '94: QuickStore: a high performance mapped object store

Seth J. White, David J. DeWitt

October 1995 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 4 Issue 4**Publisher:** Springer-Verlag New York, Inc.Full text available: [pdf\(2.58 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

QuickStore is a memory-mapped storage system for persistent C++, built on top of the EXODUS Storage Manager. QuickStore provides fast access to in-memory objects by allowing application programs to access objects via normal virtual memory pointers. This article presents the results of a detailed performance study using the OO7 benchmark. The study compares the performance of QuickStore with the latest implementation of the E programming language. The QuickStore and E systems exemplify the two ba ...

**Keywords:** benchmark, client-server, memory-mapped, object-oriented, performance, pointer swizzling**14** Distributed operating systems

Andrew S. Tanenbaum, Robbert Van Renesse

December 1985 **ACM Computing Surveys (CSUR)**, Volume 17 Issue 4**Publisher:** ACM PressFull text available: [pdf\(5.49 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Distributed operating systems have many aspects in common with centralized ones, but they also differ in certain ways. This paper is intended as an introduction to distributed operating systems, and especially to current university research about them. After a discussion of what constitutes a distributed operating system and how it is distinguished from a computer network, various key design issues are discussed. Then several examples of current research projects are examined in some detail ...

**15** Comparison of access methods for time-evolving data

Betty Salzberg, Vassilis J. Tsotras

June 1999 **ACM Computing Surveys (CSUR)**, Volume 31 Issue 2**Publisher:** ACM PressFull text available: [pdf\(529.53 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper compares different indexing techniques proposed for supporting efficient access to temporal data. The comparison is based on a collection of important performance criteria, including the space consumed, update processing, and query time for representative queries. The comparison is based on worst-case analysis, hence no

assumptions on data distribution or query frequencies are made. When a number of methods have the same asymptotic worst-case behavior, features in the methods tha ...

**Keywords:** I/O performance, access methods, structures, temporal databases

## 16 Design of the Mneme persistent object store



J. Eliot B. Moss

April 1990 **ACM Transactions on Information Systems (TOIS)**, Volume 8 Issue 2

**Publisher:** ACM Press

Full text available: pdf(3.22 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The Mneme project is an investigation of techniques for integrating programming language and database features to provide better support for cooperative, information-intensive tasks such as computer-aided software engineering. The project strategy is to implement efficient, distributed, persistent programming languages. We report here on the Mneme persistent object store, a fundamental component of the project, discussing its design and initial prototype. Mneme stores objects

## 17 Glift: Generic, efficient, random-access GPU data structures



Aaron E. Lefohn, Shubhabrata Sengupta, Joe Kniss, Robert Strzodka, John D. Owens

January 2006 **ACM Transactions on Graphics (TOG)**, Volume 25 Issue 1

**Publisher:** ACM Press

Full text available: pdf(1.52 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This article presents Glift, an abstraction and generic template library for defining complex, random-access graphics processor (GPU) data structures. Like modern CPU data structure libraries, Glift enables GPU programmers to separate algorithms from data structure definitions; thereby greatly simplifying algorithmic development and enabling reusable and interchangeable data structures. We characterize a large body of previously published GPU data structures in terms of our abstraction and prese ...

**Keywords:** Adaptive, GPGPU, GPU, adaptive shadow maps, data structures, graphics hardware, multiresolution, octree textures, parallel computation

## 18 Draft Proposed: American National Standard—Graphical Kernel System



Technical Committee X3H3 - Computer Graphics

February 1984 **ACM SIGGRAPH Computer Graphics**, Volume 18 Issue SI

**Publisher:** ACM Press

Full text available: pdf(16.07 MB)

Additional Information: [full citation](#)

## 19 Making data structures confluently persistent

Amos Fiat, Haim Kaplan

January 2001 **Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms**

**Publisher:** Society for Industrial and Applied Mathematics

Full text available: pdf(924.65 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We address a longstanding open problem of [8, 7], and present a general transformation that takes any data structure and transforms it to a confluently persistent data structure. We model this general problem using the concepts of a version DAG (Directed Acyclic Graph) and an instantiation of a version DAG. We introduce the concept of the effective depth of a vertex in the version DAG and use it to derive information theoretic lower bounds on the space expansion of any such transformation for ...

**Potpourri: Provenance management in curated databases**

Peter Buneman, Adriane Chapman, James Cheney

June 2006 **Proceedings of the 2006 ACM SIGMOD international conference on Management of data SIGMOD '06**

Publisher: ACM Press

Full text available: pdf(191.13 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Curated databases in bioinformatics and other disciplines are the result of a great deal of manual annotation, correction and transfer of data from other sources. Provenance information concerning the creation, attribution, or version history of such data is crucial for assessing its integrity and scientific value. General purpose database systems provide little support for tracking provenance, especially when data moves among databases. This paper investigates general-purpose techniques for rec ...

**Keywords:** curation, provenance, storage

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2006 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)